A Robust Collaborative Filtering Recommendation Algorithm Based on Multidimensional Trust Model

Dongyan Jia, Fuzhi Zhang, Sai Liu
School of Information Science and Engineering, Yanshan University, Qinhuangdao, China
Email: jdy_1983@163.com, xjzfz@ysu.edu.cn, liusai19@sina.com

Abstract—Collaborative filtering is one of the widely used technologies in the e-commerce recommender systems. It can predict the interests of a user based on the rating information of many other users. But the traditional collaborative filtering recommendation algorithm has the problems such as lower recommendation precision and weaker robustness. To solve these problems, in this paper we present a robust collaborative filtering recommendation algorithm based on multidimensional trust model. Firstly, according to the rating information of users, a multidimensional trust model is proposed. It measures the credibility of user’s ratings from the following three aspects: the reliability of item recommendation, the rating similarity and the user’s trustworthiness. Secondly, the computational model of trust and the traditional collaborative filtering approach are combined to select the reliable neighbor set and generate recommendation for the target user. Finally, the performances of the novel algorithm with others are compared from both sides of recommendation precision and robustness using MovieLens dataset. Compared with the existing algorithms, the proposed algorithm not only improves the quality of neighbor selection and the recommendation precision, but also has better robustness.

Index Terms—multidimensional trust model, robustness, collaborative filtering, recommender system

I. INTRODUCTION

Recommender systems, as a kind of information filtering technology, have provided an effective way to solve the information overload problem [1]. Specially, the collaborative filtering [2] is one of the most successful recommendation technologies. It generates recommendation for the target user by collecting the preference information of similar users. However, due to the sparsity of ratings, the quality of neighbor selection for target user is poor based on the similarity between users. In addition, with the emergence of shilling attacks [3,4] and the lack of credibility evaluation mechanism of ratings, how to improve the recommendation precision and robustness has become the key issue to be solved.

In order to measure the credibility of users’ ratings and the degree of trust between users, many computational models of trust have been proposed. O’Donovan et al. [5] proposed the profile-level and item-level computational model of trust and drew a conclusion that the latter performs better than the former by conducting experiments. Similarly, Lathia et al. [6] proposed an improved computational model of trust, which computed the degree of trust of target user to the recommender user based on the error of predict rating. However, both of the models generate recommendation relying on the similarity between two users. Due to the extreme sparsity of ratings, it is very difficult to compute the similarity between two users accurately, which leads to the inaccurate computation of degree of trust, poor scalability and inapplicable in large-scale dataset. Pitsilis et al. [7,8] analyzed the trust relationship between users from the point of view of subjective logic and proposed a computational model of trust based on the theory of uncertain probabilities. But the computation of uncertainty is based on the co-rated items between users. Consequently, it can’t compute the degree of trust between users accurately in the case of the extreme sparsity of ratings. Aims at the limitations of traditional collaborative filtering recommendation algorithm when selecting neighbors, Kwon et al. [9] proposed a multidimensional credibility model based on the source credibility theory [10]. They analyzed and measured the degree of trust from three aspects such as the expertise, the trustworthiness and the similarity. However, it only takes into account the heterogeneous of ratings of users and still has the vulnerability when there are attack profiles in the system.

Recently, the model-based recommendation algorithms have attracted significant attention. These algorithms use statistical methods or techniques of machine learning to construct a recommendation model of which the parameters are estimated from the rating data of users. The recommendation is generated for the target user based on the model. Jamali et al. [11] proposed a random walk model named TrustWalker, in which the predict rating for the target user on the target item would be the expected value of ratings returned by performing many random walks. But this model is greatly affected by the sparsity of ratings. Ma et al. [12] proposed a
recommendation approach based on matrix factorization named RSTE. The target user gets the recommendation by learning the latent user and item features. Moreover, they proposed the recommendation approach by incorporating social contextual information [13], and applied RSTE to the recommendation based on the implicit social relations [14]. However, this recommendation approach based on matrix factorization is greatly affected by the sparsity of direct trust information.

Aim at the problems mentioned above, on the basis of the previous work, we propose a multidimensional trust model-based robust recommendation algorithm (MTMRRA). It measures the credibility of ratings of users from different aspects. As a result, the best neighbors are selected to generate recommendation for the target user according to the computational model of trust. Our contributions are as follows.

Firstly, a multidimensional trust model is proposed, which measures the credibility of users’ ratings from the reliability of item recommendation, the rating similarity and the user’s trustworthiness based on the user-item rating matrix. So the degree of trust between users is regarded as the sum of product of each attribute and its importance weight.

Secondly, a robust collaborative filtering recommendation algorithm is presented. Based on the proposed model of trust, we can choose the best neighbors for the target user, and then get the recommendation by combining the traditional collaborative filtering recommendation approach.

Finally, we conduct the experiments on the MovieLens dataset and compare the proposed algorithm with others in terms of the MAE, RMSE and Prediction Shift metrics. Experimental results indicate that our algorithm not only improves the recommendation precision, but also has better robustness.

II. BACKGROUND

A. Description of User Rating Information

In collaborative filtering recommender systems, the rating database includes a set of \( m \) users, \( U = \{u_1,u_2,...,u_m\} \), and a set of \( n \) items, \( I = \{i_1,i_2,...,i_n\} \). Users rate some items they know with a discrete range of possible values \([\text{min},...\text{max}]\), for example, \([1,...,5]\) or \([1,...,10]\). Usually, the items with higher values are the user’s favorite ones. So the user-item rating matrix can be described as:

\[
R = \begin{bmatrix}
R_{i,1} & R_{i,2} & \cdots & R_{i,m} \\
R_{2,1} & R_{2,2} & \cdots & R_{2,n} \\
\cdots & \cdots & \cdots & \cdots \\
R_{m,1} & R_{m,2} & \cdots & R_{m,n}
\end{bmatrix},
\]

where, \( R_{i,j}(1 \leq i \leq m, 1 \leq j \leq n) \) is the rating of user \( u_i \) on item \( i_j \).

Due to the large number of items, each user often only rated a certain number of items. If user \( u_i \) hasn’t rated the item \( i_j \), we represent that as \( R_{i,j} = \emptyset \).

B. Similarity Measures

The most popular approaches of computing user similarity are cosine-based similarity and the Person correlation coefficient [15].

In cosine-based similarity approach, the ratings of each user are treated as one vector in \( n \)-dimensional space. Let the vector \( U_i \) and \( U_j \) denote the ratings of user \( u_i \) and \( u_j \) respectively, so the similarity between user \( u_i \) and \( u_j \) can be measured as:

\[
sim(u_i,u_j) = \cos(U_i,U_j) = \frac{U_i \cdot U_j}{\|U_i\|\|U_j\|}.
\]

Using Person correlation coefficient, the similarity between user \( u_i \) and \( u_j \) can be measured as:

\[
sim(u_i,u_j) = \frac{\sum_{k=1}^{n} (R_{i,k} - \overline{R_i})(R_{j,k} - \overline{R_j})}{\sqrt{\sum_{k=1}^{n} (R_{i,k} - \overline{R_i})^2 \sum_{k=1}^{n} (R_{j,k} - \overline{R_j})^2}},
\]

where \( I_p \) is the item set co-rated by user \( u_i \) and \( u_j \), and \( R_{i,k} \) and \( R_{j,k} \) are the ratings of user \( u_i \) and \( u_j \) on item \( i_k \) respectively. \( \overline{R_i} \) and \( \overline{R_j} \) are the average ratings of user \( u_i \) and \( u_j \) respectively.

C. Prediction

Assume the target user is \( u_a \), the target item is \( i_j \), so the main idea of traditional user-based collaborative filtering recommendation algorithm is as follow: based on the user-item rating matrix, the users who have rated the item \( i_j \) are selected, and the rating similarity between the target user and each of these users is computed by using the similarity measures. Then the top-\( k \) users who have larger user similarity are chosen as the neighbors of target user \( u_a \). As a result, according to the rating information of neighbors, the predict rating \( P_{a,j} \) for user \( u_a \) on item \( i_j \) is computed as:

\[
P_{a,j} = \overline{R_a} + \frac{\sum_{i \in N(u_a)} (R_{i,j} - \overline{R_i}) \cdot \sim(u_a,u_i)}{\sum_{i \in N(u_a)} \sim(u_a,u_i)}.
\]

where \( N(u_a) \) is the neighbor set of target user \( u_a \), \( \overline{R_a} \) and \( \overline{R_i} \) are the average ratings of target user \( u_a \) and the neighbor \( u_i \) respectively, \( R_{i,j} \) is the rating of \( u_i \) on item \( i_j \), \( \sim(u_a,u_i) \) is the similarity between the target user \( u_a \) and the neighbor \( u_i \).

III. MULTIDIMENSIONAL TRUST MODEL

Due to the sparsity of user-item rating matrix and the shilling attacks in collaborative filtering recommender systems, it is unreliable to select neighbors for the target user according to the similarity between users. As a result, the user’s satisfaction for the recommendation results declines. To improve the quality of selected neighbors, we propose a multidimensional trust model which analyses and measures the credibility of users’ ratings using the reliability of item recommendation, the rating similarity and the user’s trustworthiness.
A. Reliability of Item Recommendation

Definition 1. The reliability of item recommendation is defined as the degree of a user to provide an accurate prediction for every item. For user $u_k \in U$ and the item set rated by $u_k, I_k = \{i_j | R_{k,j} \neq \phi, i_j \in I\}$, the user $u_k$’s reliability of recommendation for item $i_j \in I_k$ is described as $R'_{i}$. To measure the reliability of item recommendation, we employ the item-level computational model of trust proposed by O’Donovan. Using the “leave-one-out” approach, choosing the only recommender user, every item $i_j \in I_k$ as target item, and every user $u_a$ in the user set $U = \{u_j | R_{a,j} \neq \phi, u_j \in U, u_j \neq u_k\}$ as target user, we can compute the predict rating for the target user on the target item using (3).

Based on the deviation between the predict rating and the actual rating, we can compute the user $u_k$’s reliability of recommendation for item $i_j$ as:

$$R'_{i} = \frac{\sum_{u_i \in U} \text{sat}_{ik}^i}{|U|},$$

$$\text{sat}_{ik}^i = \begin{cases} 1 & |P_{u_k,i_j} - R_{u_k,i_j}| \leq \epsilon, \\ 0 & \text{else} \end{cases},$$

where $P_{u_k,i_j}$ is the predict rating for the user $u_a$ on item $i_j$, $R_{u_k,i_j}$ is the actual rating of user $u_a$ on item $i_j$, $\epsilon$ is a threshold, we set $\epsilon=1.2$ in this paper.

B. Rating Similarity

Definition 2. Rating similarity is defined as the similarity between two users. For user $u_a \in U$ and user $u_b \in U$, the rating similarity between the two users is computed based on the item set $I_{ab} = \{i_j | R_{a,b} \neq \phi, R_{a,b} \neq \phi, i_j \in I\}$, and it is described as $S_{a,b}$.

The traditional similarity measures rely on the items co-rated between users. Due to the sparsity of user-item rating matrix, the computation of similarity has greater occasionality. To reduce its impact, we employ a relevance weighting function $f(x)$ and set a threshold for the number of co-rated items between two users by specifying the $k$ value:

$$f(x) = \frac{1}{1+e^{-x}}.$$  

(6)

The similarity $S_{a,b}$ between user $u_a$ and user $u_b$ is computed as:

$$S_{a,b} = \text{sim}(u_a, u_b) \times \frac{1}{|I|},$$

(7)

where $\text{sim}(u_a, u_b)$ is calculated according to (2), $|I|$ is the number of items co-rated by user $u_a$ and user $u_b$, $k$ is a threshold, the method of setting its value is as follows.

Let $k=1, 2, \ldots, 5$, the curve of $f(x)$ is shown in Fig. 1.

As shown in Figure 1, no matter what value the $k$ is, the $f(x)$ will be close to 1 infinitely when $x$ is greater than a certain value $x_0$. We can also get $S_{a,b} = \text{sim}(u_a, u_b)$ using (7). So we call $x_0$ the threshold of the number of co-rated items between users. Table 1 gives the comparison of value of $x_0$ when $k$ takes different values.

C. User’s Trustworthiness

Definition 3. Trustworthiness of a user is defined as the degree of his ratings that reflect the user’s actual opinions. For user $u_a \in U$ and the item set rated by $u_a$, $I_a = \{i_j | R_{a,i} \neq \phi, i_j \in I\}$, we can compute $u_a$’s trustworthiness $T_b$ using the information of the similarity of any two items in $I_b$ and the $u_a$’s ratings on the corresponding items.

Based on the user-item rating matrix, the similarity between two items is computed by using Person correlation coefficient:

$$\text{sim}(i,i_j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}) (R_{u,i_j} - \bar{R})}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R})^2 \sum_{u \in U} (R_{u,i_j} - \bar{R})^2}},$$

(8)

where $\text{sim}(i,i_j)$ is the similarity between item $i$ and item $i_j$, $U_{i}$ is the set of users who have both rated the item $i$ and item $i_j$, $U_{i,j} = \{u | R_{u,i} \neq \phi, R_{u,i_j} \neq \phi, u \in U\}$, $R_{u,i}$ and $R_{u,i_j}$ are the ratings of $u$ on item $i$ and item $i_j$ respectively, $\bar{R}_i$ and $\bar{R}_j$ are the average rating of item $i$ and $i_j$ respectively.

We can get a real value in the range $[-1, +1]$ from (8), which is mapped to the range $[0, 1]$ by using $\text{sim}(i,i_j) = \frac{1 + \text{sim}(i,i_j)}{2}$.

Consequently, the user $u_a$’s trustworthiness $T_b$ is computed as:

![Figure 1](image-url)
where $t_{ij}^b$ is the trustworthiness of $u_b$ for item pair $(i,j)$, $\text{sim}(i,j)$ is the similarity between item $i$ and item $j$, $R_{uj}$ and $R_{uj}$ are the ratings of $u_b$ on item $i$ and item $j$ respectively.

D. Computation of Trust Degree

Based on the analysis above, we can compute the degree of trust of user $u_a$ to user $u_b$ as:

$$
\text{trust}_{a,b} = \alpha R_u^i + \beta S_{a,b} + \gamma T_b,
$$

where $R_u^i$ is the reliability of item recommendation of user $u_b$ for item $i$, $S_{a,b}$ is the rating similarity between user $u_a$ and user $u_b$, $T_b$ is the trustworthiness of user $u_b$, $\alpha$, $\beta$, $\gamma$ are the importance weights of each attribute above, we can set their values according to the following method.

Using the experimental dataset, we can simulate the performance of four recommendation strategies as follows:

- **CF** - Traditional user-based collaborative filtering recommendation strategy.
- **RCF** - Reliability of item recommendation-based collaborative filtering recommendation strategy.
- **SCF** - Rating similarity-based collaborative filtering recommendation strategy.
- **TCF** - User’s trustworthiness-based collaborative filtering recommendation strategy.

With the number of neighbors increasing, we can get the MAE values of each recommendation strategy respectively. Compared with the recommendation precision of CF, the percentage of improvement for each recommendation strategy is calculated. Let $p_{ref}$, $p_{ref}$ and $p_{ref}$ be the percentage of improvement for recommendation strategy of RCF, SCF and TCF respectively, so we have:

$$
\alpha = \frac{p_{ref}}{p_{ref} + p_{ref} + p_{ref}},
$$
$$
\beta = \frac{p_{ref}}{p_{ref} + p_{ref} + p_{ref}},
$$
$$
\gamma = \frac{p_{ref}}{p_{ref} + p_{ref} + p_{ref}}.
$$

Obviously, the greater percentage of improvement the recommendation strategy has, the larger the corresponding importance weight is. Considering the values of $\alpha$, $\beta$ and $\gamma$ are different using different dataset, so their values should be computed dynamically.

Let's take an example to illustrate the computational process of the values of $\alpha$, $\beta$ and $\gamma$. Using the MovieLens1 dataset, we randomly select 754 users’ profiles as the training set and the remaining as the test set to conduct the experiments and compare the performances of the RCF, SCF and TCF with CF. Table 2 shows the comparison of recommendation precision with different number of neighbors.

### TABLE II. COMPARISON OF RECOMMENDATION PRECISION (MAE)

<table>
<thead>
<tr>
<th>number of neighbors</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>CF</td>
<td>0.7254</td>
<td>0.7331</td>
<td>0.7287</td>
<td>0.7235</td>
<td>0.7091</td>
<td>0.7061</td>
<td>0.7096</td>
<td>0.7094</td>
<td>0.7045</td>
<td>0.7010</td>
</tr>
<tr>
<td>RCF</td>
<td>0.6309</td>
<td>0.6478</td>
<td>0.6506</td>
<td>0.6681</td>
<td>0.6676</td>
<td>0.6527</td>
<td>0.6600</td>
<td>0.6610</td>
<td>0.6648</td>
<td>0.6585</td>
</tr>
<tr>
<td>SCF</td>
<td>0.7276</td>
<td>0.7324</td>
<td>0.7275</td>
<td>0.7203</td>
<td>0.7055</td>
<td>0.7066</td>
<td>0.7071</td>
<td>0.7048</td>
<td>0.7034</td>
<td>0.6998</td>
</tr>
<tr>
<td>TCF</td>
<td>0.7008</td>
<td>0.6911</td>
<td>0.6964</td>
<td>0.6805</td>
<td>0.6860</td>
<td>0.6825</td>
<td>0.6841</td>
<td>0.6801</td>
<td>0.6892</td>
<td>0.6857</td>
</tr>
</tbody>
</table>

As shown in Table 2, the recommendation strategy of RCF, SCF and TCF all outperform the CF in terms of recommendation precision. Compared with the CF strategy, the average percentage of improvement for the RCF, SCF and TCF is 8.06%, 0.16% and 3.76% respectively. So:

$$
\alpha = \frac{8.06\%}{8.06\% + 0.16\% + 3.76\%} = 0.6728,
$$
$$
\beta = \frac{0.16\%}{8.06\% + 0.16\% + 3.76\%} = 0.0134,
$$
$$
\gamma = \frac{3.76\%}{8.06\% + 0.16\% + 3.76\%} = 0.3139.
$$

IV. MULTIDIMENSIONAL TRUST MODEL-BASED ROBUST RECOMMENDATION ALGORITHM

A. Description of Algorithm

To improve the recommendation precision, a multidimensional trust model-based robust recommendation algorithm (MTMRRA) is proposed. The main steps of MTMRRA algorithm are described as follows: according to the rating information of users, select the user set $C(u_b)$ who have rated the target item $i_j$ and use (11) to compute the degree of trust of target user $u_a$ to each user in $C(u_b)$. Based on that, select the top-$k$ users as the neighbors of target user $u_a$ and compute the predicted rating $P_{u_a}$ for the target user $u_a$ on the target item $i_j$ as:

\[1\text{http://www.grouplens.org/node/73]
The algorithm is described as follows.

**Algorithm:** MTMRRA

**Input:** the user-item rating matrix \(R\)

**Output:** the predicted rating \(P_{ui}\) for target user \(u_{i}\) on the target item \(j\).

**Begin**

1: \(C(u_{i}) \leftarrow \{u_{j} | R_{i,j} \neq \phi, u_{j} \in U\}\)

2: for each \(u_{j} \in C(u_{i})\) do

3: \(I_{i} \leftarrow \{i_{j} | R_{j,i} \neq \phi, j \in I\}\)

4: \(U_{i} \leftarrow \{u_{j} | R_{i,j} \neq \phi, u_{j} \in U, u_{j} \neq u_{i}\}\)

5: \(\text{sum}_\text{satisfactory} \leftarrow 0\)

6: for each \(u_{j} \in U_{i}\) do

7: \(P_{uj} \leftarrow \text{Predict}(u_{j}, u_{i}, i)\)

8: \(\text{sat}_{ij} \leftarrow \text{Satisfactory}(u_{j}, u_{i}, i)\)

9: \(\text{sum}_\text{satisfactory} \leftarrow \text{sum}_\text{satisfactory} + \text{sat}_{ij}\)

10: end for

11: \(R_{ij}^l \leftarrow \frac{\text{sum}_\text{satisfactory}}{|U_{i}|}\)

12: \(I_{i} \leftarrow \{i_{j} | R_{j,i} \neq \phi, R_{j,i} \neq \phi, j \in I\}\)

13: \(\text{sim}(u_{i}, u_{a}) \leftarrow \text{similarity}(u_{i}, u_{a})\)

14: \(S_{a,k} \leftarrow \text{sim}(u_{i}, u_{a}) \times f(|I_{a}|)\)

15: \(\text{sum}_\text{trustworthy} \leftarrow 0\)

16: for \(\forall i_{k}, i_{j} \in I_{i} (i_{j} \neq i_{j})\) do

17: \(\text{sim}(i_{k}, i_{j}) \leftarrow \text{similarity}(i_{k}, i_{j})\)

18: \(t_{k,j} \leftarrow \text{trustworthy}(u_{i}, i_{k}, i_{j})\)

19: \(\text{sum}_\text{trustworthy} \leftarrow \text{sum}_\text{trustworthy} + t_{k,j}\)

20: end for

21: \(T_{i} \leftarrow \frac{2 \times \text{sum}_\text{trustworthy}}{|I_{i}| - 1}\)

22: \(\text{trust}_{ij} \leftarrow \alpha R_{ij}^l + \beta S_{a,k} + \gamma T_{i}\)

23: end for

24: \(N(u_{i}) \leftarrow \{u_{j} | S_{a,k} > 0, \text{trust}_{ij} > T, u_{j} \in C(u_{i})\}\)

25: sort the degree of trust of the target user \(u_{i}\) to every user in the \(N(u_{i})\);

26: \(U^{+} \leftarrow \{u_{j} | u_{j} \in N(u_{i}), i = 1, 2, \ldots, k\}\)

27: \(P_{ui} \leftarrow \text{Predict}_\text{MTMRRA}(u_{i}, i)\)

28: return \(P_{ui}\)

**End**

This algorithm consists of three parts. The first part, the first line, is to get the user set \(C(u_{i})\) who have rated the target item. The second part, from line 2 to 23, is to compute the degree of trust of the target user to each user in \(C(u_{i})\). The third part, from line 24 to 28, is to select the neighbors for the target user and compute the predicted rating \(P_{ui}\) for target user \(u_{i}\) on the target item \(i\).

**B. Complexity Analysis**

In the process of computing the degree of trust of target user \(u_{i}\) to each user in \(C(u_{i})\), the computation of the reliability of item recommendation, the rating similarity and the user’s trustworthiness is of complexity \(O(m)\), \(O(l)\) and \(O(l^{2})\) respectively, where \(m\) denotes the number of users in the recommender system, \(l\) denotes the number of ratings rated by one user. In the actual recommender system, the degree of trust between users is usually computed off-line, so its complexity is \(O(1)\). The complexity of selecting neighbors and computing the predict ratings for the target user is \(O(m^{2})\) and \(O(k)\) respectively, where \(k\) denotes the number of neighbors. So the total complexity of computation online is \(O(m^{2}+k)\). Considering the fact that we often have \(k<<m\), so the complexity of the algorithm MTMRRA is \(O(m^{2})\).

**C. Proof of Correctness**

The correctness of the algorithm MTMRRA is proved from both sides of theoretical study and experimental analysis. Theoretically, the correctness of MTMRRA depends on the correct computation of the degree of trust of target user to others. The computational approach of trust proposed in this paper measures the credibility of user’s ratings from three aspects, which not only considers the potential relationship between user profiles, but also evaluates the credibility of user profile itself from both sides of horizontal and vertical. The algorithm MTMRRA makes a multi-dimensional analysis for the degree of trust between users, which is feasible and correct in theory. From the experiment point of view, the algorithm is correct in case that compared to other recommendation algorithms, it not only improves the recommendation precision, but also has better robustness. We can see that the MTMRRA is correct obviously through section VI.
computed by measuring the deviation between the prediction rating and the actual rating. Obviously, the smaller MAE (or RMSE) is, the higher the recommendation precision of algorithm is. MAE and RMSE can be computed as:

\[
\text{MAE} = \frac{1}{n} \sum_{j=1}^{n} |p_j - r_j|, \quad (14)
\]

\[
\text{RMSE} = \sqrt{\frac{1}{n} \sum_{j=1}^{n} (p_j - r_j)^2}, \quad (15)
\]

where \(p_j\) is the predicted rating given to the target user on target item \(i_j\), \(r_j\) is the actual rating of the target user on target item \(i_j\), \(n\) is the total number of prediction.

Moreover, we use MAE, RMSE and Prediction Shift to evaluate the robustness of algorithm. The Prediction Shift measures the deviation of prediction on the attacked item (before and after attack) of the recommendation algorithm. The smaller the Prediction Shift is, the better robustness the algorithm has. It is computed as:

\[
\text{PredShift}(u,i) = \frac{1}{n} \sum_{j=1}^{n} p'(u,i_j) - p(u,i_j), \quad (16)
\]

where \(p'(u,i_j)\) and \(p(u,i_j)\) are the predicted ratings for user \(u_k\) on item \(i_j\) before and after the item \(i_j\) is attacked, \(n\) is the total number of prediction.

C. Recommendation Precision Analysis

To evaluate the recommendation precision of algorithms, we have carried out the experiments with our multidimensional trust model-based robust recommendation algorithm (MTMRRA), the traditional collaborative filtering recommendation algorithm (CF) and the recommendation algorithm proposed by O’Donovan. With the number of neighbors increasing gradually, Fig. 2 gives the comparisons of recommendation precision.

As shown in Fig. 2, the MTMRRA algorithm outperforms the CF algorithm and the O’Donovan’s in term of recommendation precision. Compared with CF algorithm, MTMRRA algorithm improves by 9.18% (MAE) and 8.95% (RMSE); compared with O’Donovan’s algorithm, MTMRRA algorithm improves by 6.39% (MAE) and 6.01% (RMSE). That is to say that it is helpful to improve the quality of chosen neighbors and the recommendation precision by measuring the credibility of users’ ratings from several aspects. Therefore, our MTMRRA algorithm has better recommendation precision.

D. Robustness Analysis

To evaluate the robustness of the algorithms, we inject some hybrid attack profiles (the same number of profiles of random attack, average attack and bandwagon attack) into the training set. Let the filler size be 1%, 3%, 5%, 10%, 25%, the attack size be 1%, 2%, 5%, 10%, and the number of neighbors be 40, with the filler size and attack size increasing gradually, the comparisons of recommendation precision of algorithm MTMRRA, CF and O’Donovan’s are shown in Table 3 and Table 4.

| Table III. Comparison of MAE Values |
|-------------------------------|----------------|----------------|----------------|----------------|----------------|----------------|
| attack size | 1% | 2% | 5% | 10% |
| algorithm | CF | O’Donovan | MTMRRA | CF | O’Donovan | MTMRRA | CF | O’Donovan | MTMRRA | CF | O’Donovan | MTMRRA |
| filler size=1% | 0.7374 | 0.7028 | 0.6705 | 0.7359 | 0.7041 | 0.6770 | 0.7478 | 0.7057 | 0.6446 | 0.7642 | 0.7122 | 0.6783 |
| filler size=3% | 0.7337 | 0.6984 | 0.6755 | 0.7411 | 0.7046 | 0.6730 | 0.7441 | 0.7061 | 0.6717 | 0.7534 | 0.7104 | 0.6964 |
| filler size=5% | 0.7371 | 0.7015 | 0.6760 | 0.7409 | 0.7056 | 0.6707 | 0.7541 | 0.7303 | 0.6694 | 0.7651 | 0.7405 | 0.6918 |
| filler size=10% | 0.7340 | 0.7078 | 0.6596 | 0.7454 | 0.7143 | 0.6619 | 0.7550 | 0.7093 | 0.6780 | 0.7458 | 0.7092 | 0.6672 |
| filler size=25% | 0.7219 | 0.6963 | 0.6510 | 0.7488 | 0.7106 | 0.6563 | 0.7309 | 0.6976 | 0.6686 | 0.7381 | 0.7122 | 0.6869 |
TABLE IV. COMPARISON OF RMSE VALUES

<table>
<thead>
<tr>
<th>attack size</th>
<th>CF</th>
<th>O’Donovan</th>
<th>MTMRRA</th>
<th>CF</th>
<th>O’Donovan</th>
<th>MTMRRA</th>
<th>CF</th>
<th>O’Donovan</th>
<th>MTMRRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1%</td>
<td>0.9130</td>
<td>0.8823</td>
<td>0.8417</td>
<td>0.9143</td>
<td>0.8805</td>
<td>0.8494</td>
<td>0.9324</td>
<td>0.8814</td>
<td>0.8113</td>
</tr>
<tr>
<td>2%</td>
<td>0.9135</td>
<td>0.8790</td>
<td>0.8480</td>
<td>0.9205</td>
<td>0.8806</td>
<td>0.8482</td>
<td>0.9248</td>
<td>0.8790</td>
<td>0.8400</td>
</tr>
<tr>
<td>5%</td>
<td>0.9139</td>
<td>0.8821</td>
<td>0.8497</td>
<td>0.9137</td>
<td>0.8812</td>
<td>0.8494</td>
<td>0.9378</td>
<td>0.9053</td>
<td>0.8384</td>
</tr>
<tr>
<td>10%</td>
<td>0.9070</td>
<td>0.8775</td>
<td>0.8224</td>
<td>0.9107</td>
<td>0.8802</td>
<td>0.8213</td>
<td>0.9157</td>
<td>0.8700</td>
<td>0.8475</td>
</tr>
<tr>
<td>1%</td>
<td>0.8959</td>
<td>0.8685</td>
<td>0.8221</td>
<td>0.9183</td>
<td>0.8820</td>
<td>0.8261</td>
<td>0.8877</td>
<td>0.8595</td>
<td>0.8245</td>
</tr>
<tr>
<td>3%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As shown in Table 3 and Table 4, the MTMRRA algorithm outperforms the CF algorithm and the O’Donovan’s algorithm in terms of recommendation precision whatever the attack size is. Also, with the attack size increasing gradually, the recommendation precision of three algorithms follows to descend. It is now clear that the more attack users in the system, the lower quality of recommendation. Compared with the recommendation precision of CF, MTMRRA algorithm improves by 9.74% (MAE) and 8.56% (RMSE); compared with the recommendation precision of O’Donovan’s, MTMRRA algorithm improves by 5.32% (MAE) and 4.6% (RMSE). It can be proved that our MTMRRA algorithm has better robustness.

As shown in Fig. 3, under the same filler size, with the attack size increasing gradually, the prediction shift of all algorithms increases. So the more attack users in the system, the lower quality of recommendation. In addition, under the same attack size and filler size, the MTMRRA algorithm outperforms the CF and O’Donovan’s in terms of prediction shift.

VII CONCLUSIONS

With the wide application of the collaborative filtering algorithm in e-commerce, how to improve the recommendation precision and the robustness becomes more and more important. We have made some beneficial explorations in this area. In this paper we propose a multidimensional trust model which measures the credibility of users’ ratings from three attributes. Based on the model of trust, we present a robust collaborative filtering recommendation algorithm. Compared with other algorithms, the proposed algorithm not only improves the recommendation precision, but also has better robustness. The experiments conducted on MovieLens prove the effectiveness of the proposed
algorithm. With the new ratings adding into the system gradually, how to design an incremental recommendation algorithm and generate recommendation for the target user accurately will be our future work.

ACKNOWLEDGMENT

The work described in this paper was supported by the Natural Science Foundation of Hebei Province, China (No. F2011203219).

REFERENCES


Dongyan Jia was born in Hebei Province, China in 1983. She received her M.S. in Computer Application Technology from Yanshan University, China, in 2009. She is currently a Ph.D. Candidate in the School of Information Science and Engineering, Yanshan University, Qinhuangdao, China. Her research interests include collaborative filtering, trusted computing and information security.

Fuzhi Zhang was born in Henan Province, China in 1964. He received his Ph.D. in Computer Application Technology from Beijing Institute of Technology, China, in 2003. He is currently a professor in the School of Information Science and Engineering, Yanshan University, Qinhuangdao, China. His research domains cover intelligent network information processing, network and information security, service-oriented computing.

Sai Liu was born in Henan Province, China in 1987. He received his B.S. in Computer Application Technology from Zhengzhou University, China, in 2010. He is currently a M.S. Candidate in the School of Information Science and Engineering, Yanshan University, Qinhuangdao, China. His research interests include collaborative filtering and contextual recommendation.